

IN THE UNITED STATES PATENT AND TRADEMARK OFFICE

Patent Application

Applicant(s): D. A. Brown et al.

Case: 8-1

Serial No.: 10/699,092

Filing Date: October 31, 2003

Group: 2457

Examiner: Hussein A. El Chanti

Title: Internal Memory Controller Providing Configurable Access
of Processor Clients to Memory Instances

AMENDED APPEAL BRIEF

Commissioner for Patents
P.O. Box 1450
Alexandria, VA 22313-1450

Sir:

This Amended Appeal Brief is being filed in response to the Notification of Non-Compliant Appeal Brief dated January 7, 2009. In the objected-to Appeal Brief, dated October 21, 2008, Applicants (hereinafter “Appellants”) appealed the final rejection, dated June 20, 2008, of claims 1-16 and 20 of the above-identified application. Appellants respectfully traverse the objections raised to the Appeal Brief, on the grounds that the summary of claimed subject matter contained in the Appeal Brief was in fact compliant with 37 C.F.R. §41.37(c)(1)(v). Notwithstanding this traversal, Appellants submit the present Amended Appeal Brief solely to conform to the subjective preferences indicated in the January 7, 2009, Notification of Non-Compliant Appeal Brief.

REAL PARTY IN INTEREST

The present application is assigned of record to Agere Systems Inc. On April 2, 2007, the assignee Agere Systems Inc. completed a merger with LSI Logic Corporation, with the resulting entity being named LSI Corporation. LSI Corporation is the real party in interest.

RELATED APPEALS AND INTERFERENCES

There are no known related appeals or interferences.

STATUS OF CLAIMS

The present application was filed on October 31, 2003 with claims 1-20. Claims 17-19 were previously canceled. Claims 1-16 and 20 remain pending, with claims 1 and 20 being the pending independent claims.

Each of claims 1-16 and 20 stands rejected under 35 U.S.C. §103(a). Claims 1-16 and 20 are appealed.

STATUS OF AMENDMENTS

There have been no amendments filed subsequent to the final rejection. In an Advisory Action dated September 9, 2008, the Examiner withdrew an objection to the drawings.

SUMMARY OF CLAIMED SUBJECT MATTER

Independent claim 1 recites a network processor integrated circuit that comprises a plurality of processor clients internal to the network processor integrated circuit, an internal memory having a plurality of memory instances, and an internal memory controller for controlling access of the plurality of processor clients to the plurality of memory instances. The internal memory controller comprises a configurable switching element, with the configurable switching element being connectable between the plurality of processor clients and the plurality of memory instances and being operative to control access of particular ones of the plurality of processor clients to particular ones of the plurality of memory instances. The configurable switching element is configurable to connect any one of at least a subset of the plurality of

processor clients to each of at least a subset of the plurality of memory instances. In a first selectable configuration of the configurable switching element, a given one of the processor clients is permitted to access a first set of memory instances comprising one or more of the plurality of memory instances, and in a second selectable configuration of the configurable switching element, the given processor client is permitted to access a second set of memory instances comprising one or more of the plurality of memory instances, the second set being different than the first set.

In an illustrative embodiment described in the specification at, for example, page 4, line 19, to page 5, line 12, with reference to FIG. 1, a network processor integrated circuit (e.g., 102 in FIG. 1) that comprises a plurality of processor clients (e.g., 120 in FIG. 1) internal to the network processor integrated circuit, an internal memory (e.g., 104 in FIG. 1) having a plurality of memory instances (e.g., 122 in FIG. 1), and an internal memory controller (e.g., 115 in FIG. 1) for controlling access of the plurality of processor clients to the plurality of memory instances. As described in the specification at, for example, page 5, lines 18-22, with reference to FIG. 1, and at page 7, line 22, to page 8, line 2, with reference to FIG. 2, the internal memory controller comprises a configurable switching element (e.g., 200 in FIG. 2), with the configurable switching element being connectable between the plurality of processor clients and the plurality of memory instances and being operative to control access of particular ones of the plurality of processor clients to particular ones of the plurality of memory instances. As described in the specification at, for example, page 5, lines 22-25, the configurable switching element is configurable to connect any one of at least a subset of the plurality of processor clients to each of at least a subset of the plurality of memory instances. As described in the specification at, for example, page 5, line 27, to page 6, line 4, in a first selectable configuration of the configurable switching element, a given one of the processor clients is permitted to access a first set of memory instances comprising one or more of the plurality of memory instances, and in a second selectable configuration of the configurable switching element, the given processor client is permitted to access a second set of memory instances comprising one or more of the plurality of memory instances, the second set being different than the first set.

Dependent claim 11 recites the network processor integrated circuit of claim 1 wherein addresses are allocated to multiple memory instances associated with the given processor client in order of decreasing memory instance size.

In an illustrative embodiment described in the specification at page 11, lines 9-12, with reference to FIG. 4A, addresses are allocated to the memory instances in order of decreasing memory instance size. More specifically, the biggest memory instance is allocated first, that is, assigned addresses starting with 00000h in this example, where “h” denotes hexadecimal notation, while the smallest memory instance is allocated last.

Dependent claim 12 recites the network processor integrated circuit of claim 1 wherein multiple memory instances associated with the given processor client have different sizes which are related to one another as multiples of two.

In an illustrative embodiment described in the specification at, for example, page 10, line 24, to page 11, line 4, the sizes of the internal memory instances can vary from 32 KB to 1 MB in multiples of two. With reference to the table in FIG. 4A, a given client in this example may be assigned the nine memory instances denoted mem1, mem2, . . . mem9. Mem1 and mem2 each have a size of 1 MB, mem3 and mem4 each have a size of 512 KB, mem5 has a size of 256 KB, mem6 and mem7 each have a size of 128 KB, mem8 has a size of 64 KB, and mem9 has a size of 32 KB.

Dependent claim 13 recites the network processor integrated circuit of claim 1 wherein a different set of mask bits is associated with each of a plurality of different memory instance sizes, and a different address decoder value is associated with each of the plurality of memory instances.

In an illustrative embodiment described in the specification at, for example, page 11, lines 12-19, with reference to FIG. 4B, a different set of mask bits is associated with each of the different memory instance sizes. Specifically, the table of FIG. 4B lists the memory sizes and the particular set of mask bits associated with each memory size. In this example, which utilizes a 20-bit address denoted as bits [19:0], the mask bits comprise a seven-bit value denoted as bits [6:0]. Thus, the mask bits associated with the 1 MB memory instance size are 0011111, the

mask bits associated with the 512 KB memory instance size are 0001111, the mask bits associated with the 256 KB memory instance size are 0000111, and so on.

In an illustrative embodiment described in the specification at, for example, page 11, lines 20-27, with reference to FIG. 4A, a different address decoder value is associated with each of the nine memory instances mem1 through mem9. Once the memory instances for the given client are organized in decreasing size as described above with regard to claim 11, with the biggest memory instance starting with the 0th address, the upper seven bits of the memory instance start address are used as the address decoder value for the biggest memory instance. So in the FIG. 4A example, mem1 has start address 00000h so the corresponding address decoder value is the seven-bit value 0000000b, where “b” denotes binary notation. For mem2 the start address is 40000h, so its corresponding address decoder value is the seven-bit value 0100000b. For mem9 the start address is E4000h, so its corresponding address decoder value is the seven-bit value 1110010b.

Dependent claim 14 recites the network processor integrated circuit of claim 13 wherein address decoding logic applies the mask bits for a given memory instance to an incoming address from the given processor client, and compares the result to the address decoder value for the given memory instance to determine if the incoming address is directed to an address in the given memory instance.

In an illustrative embodiment described in the specification at, for example, page 12, lines 3-12, the address decoding logic applies the mask bits to the upper seven bits of an incoming address, and compares the result to an address decoder value stored in a configuration register to see if there is an address hit for a particular one of the memory instances. In the event of an address hit, the incoming address can be passed as is to the appropriate memory instance.

Dependent claim 15 recites the network processor integrated circuit of claim 14 wherein a decoded address is considered valid for the given processor client only if a master client identifier stored for the given memory instance specifies the given processor client.

In an illustrative embodiment described in the specification at, for example, page 13, lines 8-22, a stored five-bit value identifies a “master” client for each memory instance. In this example, it is assumed that there are no more than 20 distinct clients, denoted CL1 through

CL20, with each client having a particular five-bit identifier, namely 00001 for CL1, . . . , 10100 for CL20. The master client identifier specifies which of the clients, if any, has been assigned to the particular memory instance. This allows the address decoder logic to determine the appropriate client whose access request address is validly decoded for the particular memory instance, so that any other such access requests from other clients that are directed to the particular memory instance can be ignored. Thus, a decoded address is considered valid for a given client only if the master client identifier stored for the particular memory instance specifies the given client. A value of 00000 in this implementation indicates that there is no master client for the particular memory instance.

Independent claim 20 recites a method for use in a network processor integrated circuit for controlling access of a plurality of processor clients internal to the network processor integrated circuit to a plurality of memory instances of an internal memory of the network processor integrated circuit. The method includes a step of providing within the network processor integrated circuit an internal memory controller comprising a configurable switching element. The configurable switching element is connectable between the plurality of processor clients and the plurality of memory instances and being operative to control access of particular ones of the plurality of processor clients to particular ones of the plurality of memory instances. The configurable switching element is configurable to connect any one of at least a subset of the plurality of processor clients to each of at least a subset of the plurality of memory instances. The method also includes a step of selecting one of at least a first selectable configuration and a second selectable configuration of the configurable switching element, wherein in the first selectable configuration a given one of the processor clients is permitted to access a first set of memory instances comprising one or more of the plurality of memory instances, and in the second selectable configuration the given processor client is permitted to access a second set of memory instances comprising one or more of the plurality of memory instances, the second set being different than the first set.

As described in the specification at, for example, page 4, line 19, to page 5, line 12, with reference to FIG. 1, an illustrative embodiment includes a method for use in a network processor integrated circuit (e.g., 102 in FIG. 1) for controlling access of a plurality of processor clients

(e.g., 120 in FIG. 1) internal to the network processor integrated circuit to a plurality of memory instances (e.g., 122 in FIG. 1) of an internal memory (e.g., 104 in FIG. 1) of the network processor integrated circuit. As described in the specification at, for example, page 5, lines 18-22, with reference to FIG. 1, and at page 7, line 22, to page 8, line 2, with reference to FIG. 2, the method includes a step of providing within the network processor integrated circuit an internal memory controller (e.g., 115 in FIG. 1) comprising a configurable switching element (e.g., 200 in FIG. 2). As described in the specification at, for example, page 5, lines 18-22, the configurable switching element is connectable between the plurality of processor clients and the plurality of memory instances and being operative to control access of particular ones of the plurality of processor clients to particular ones of the plurality of memory instances. As described in the specification at, for example, page 5, lines 22-25, the configurable switching element is configurable to connect any one of at least a subset of the plurality of processor clients to each of at least a subset of the plurality of memory instances. As described in the specification at, for example, page 5, line 27, to page 6, line 4, the method also includes a step of selecting one of at least a first selectable configuration and a second selectable configuration of the configurable switching element, wherein in the first selectable configuration a given one of the processor clients is permitted to access a first set of memory instances comprising one or more of the plurality of memory instances, and in the second selectable configuration the given processor client is permitted to access a second set of memory instances comprising one or more of the plurality of memory instances, the second set being different than the first set.

The claimed invention provides a number of significant advantages over conventional arrangements. As described in the specification at, for example, page 3, lines 6-13, the techniques of the invention in an illustrative embodiment allow each of a number of processor clients to be connected to any of a number of memory instances, in a fully configurable manner. This significantly improves internal memory bandwidth and throughput performance of a network processor. Moreover, a single network processor configured in accordance with the invention can be adapted for use in a variety of processing applications having different memory access requirements. Also, support for multiple assignments of memory instances to clients may

be provided using a minimum amount of internal memory, thereby reducing processor size, cost and power consumption.

GROUND OF REJECTION TO BE REVIEWED ON APPEAL

Claims 1-16 and 20 are rejected under 35 U.S.C. §103(a) as being unpatentable over U.S. Patent Application Publication No. 2004/0049564 (hereinafter “Ng”) in view of U.S. Patent No. 6,665,268 (hereinafter “Sato”).

ARGUMENT

Claims 1-10, 16 and 20

As discussed above, independent claim 1 recites a network processor integrated circuit that comprises a plurality of processor clients internal to the network processor integrated circuit, an internal memory having a plurality of memory instances, and an internal memory controller for controlling access of the plurality of processor clients to the plurality of memory instances. The claim further recites that the internal memory controller comprises a configurable switching element, with the configurable switching element being connectable between the plurality of processor clients and the plurality of memory instances and being operative to control access of particular ones of the plurality of processor clients to particular ones of the plurality of memory instances. The configurable switching element is configurable to connect any one of at least a subset of the plurality of processor clients to each of at least a subset of the plurality of memory instances. In a first selectable configuration of the configurable switching element, a given one of the processor clients is permitted to access a first set of memory instances comprising one or more of the plurality of memory instances, and in a second selectable configuration of the configurable switching element, the given processor client is permitted to access a second set of memory instances comprising one or more of the plurality of memory instances, the second set being different than the first set.

It is important to note that the claim expressly recites that the plurality of processor clients, the plurality of memory instances, and the internal memory controller with configurable

switching element having first and second selectable configurations are all within the recited network processor integrated circuit itself.

An illustrative embodiment of the claimed arrangement is shown in FIG. 1 of the drawings. There, a network processor 102 comprises processor clients 120, memory instances 122 of an internal memory 104, and a memory controller 115. As in the above-noted limitations of claim 1, the processor clients 120, the memory instances 122 and the memory controller 115 are all internal to the network processor 102. See the specification at, for example, page 5, lines 7-12. The specification at page 1, lines 10-13 indicates that a network processor generally controls the flow of packets between a physical transmission medium and a switch fabric in a router of other type of packet switch. The specification further makes clear that the above-described illustrative network processor 102 may be configured as an integrated circuit to provide an interface between a network and a switch fabric in a router or switch. See the specification at, for example, page 14, lines 12-13.

The collective teachings of Ng and Sato clearly fail to disclose or suggest a network processor integrated circuit having as internal elements thereof the recited processor clients, memory instances and memory controller with configurable switching element having first and second selectable configurations as recited.

The Examiner in formulating the §103(a) rejection of claim1 fails to address the network processor integrated circuit limitations of the claim. Instead, the Examiner refers to claim 1 as simply “an apparatus.” See the final Office Action at page 2, last two lines, and page 3, first two lines. However, what is claimed is not an apparatus generally, but a particular type of apparatus, namely a network processor integrated circuit. As noted above, the specification indicates that a network processor is a particular type of processor, one that generally controls the flow of packets between a physical transmission medium and a switch fabric. Neither Ng nor Sato disclose such a network processor implemented as an integrated circuit and having as internal elements of said integrated circuit the recited processor clients, memory instances and memory controller with configurable switching element having first and second selectable configurations.

The Examiner relies primarily on the arrangement shown in FIG. 2 of Ng and the corresponding text in paragraphs [0036] through [0039]. However, this figure shows a storage

area network (SAN) that connects multiple server devices 202 with multiple storage devices 230-238. It is clear from, for example, paragraph [0001] of Ng that a given SAN and its associated server devices and storage devices are not elements of a network processor integrated circuit.

The Examiner apparently acknowledges these fundamental deficiencies of Ng as applied to claim 1, but argues that any missing teachings can be found in Sato in FIGS. 9A and 9B and columns 25 and 26. See the final Office Action at page 4, first paragraph. However, the crossbar network unit 10 as shown in the relied-upon figures is part of “a parallel processor system,” further described in Sato as “a kind of supercomputer,” and not part of a network processor integrated circuit of the type recited in the claim. See Sato at column 11, lines 48-57.

Moreover, the crossbar network unit 10 is operative to interconnect an arbitrary set of the processor elements of the parallel processor system with one another for purposes of load testing the parallel processor system. See Sato at, for example, column 12, lines 15-29, column 12, line 56, to column 13, line 10. Accordingly, the crossbar network unit 10 is not an element of a network processor integrated circuit, nor is it configurable to connect processor clients of such a network processor integrated circuit to internal memory instances of such a network processor integrated circuit, as recited.

Indeed, Sato contains numerous indications that the processor elements are physically distinct from the crossbar network unit, as well as from each other. For example, Sato at column 1, lines 30-35, states that the “parallel processor system is generally called the supercomputer, in which an ultrahigh speed arithmetic operation is realized by parallel processing of a plurality of processor elements interconnected through an inter-processor network (such as a crossbar network unit).” As such, crossbar network unit 10 is an inter-processor network which interconnects a plurality of processor elements, rather than a component of a network processor integrated circuit.

It is therefore clear that the collective teachings of Ng and Sato fail to meet the limitations of claim 1.

The Examiner further argues that it would be obvious to combine Ng with Sato to meet the claimed arrangement. However, as indicated above, the relied-upon portions of Sato teach the use of a crossbar network unit 10 for interconnecting processor elements of a parallel

processor system with one another. The relied-upon portions of Ng, on the other hand, disclose a SAN that connects multiple server devices 202 with multiple storage devices 230-238. It is not clear why one skilled in the art would be motivated to adapt the crossbar network unit 10, utilized in Sato to interconnect supercomputer processor elements with one another for purposes of load testing, to communication between server devices and storage devices in a SAN of the type disclosed in Ng.

The Examiner argues that such a combination would be obvious because it “would decrease the delays in communication between the client and the memory . . . caused by physical distance between the devices.” See the final Office Action at page 4, first paragraph. However, the portion of Sato at column 2, lines 38-43, cited in support of this alleged motivation involves a parallel processor system or supercomputer in which the various processor elements may be separated from one another by large physical distances.

More generally, Sato is directed to techniques for improving a load test of a parallel processing system by considering differences in the time required for a source processing element to transmit a packet to the crossbar network unit; these differences in transmission time are caused by differences in the physical distance between the crossbar network and the processor elements. However, Sato does not teach or suggest reducing these physical distances in order to reduce these transmission times; rather, Sato teaches using the differences in transmission times in order to transmit packets from different processing elements such that these packets will arrive simultaneously at the crossbar network unit and thus maximize the load on the crossbar network unit. See Sato at, for example, column 2, lines 16-43; column 3, lines 5-23; column 25, lines 29-35; column 28, lines 61-67; column 29, line 47, to column 30, line 53.

The present invention, by way of contrast, involves a network processor integrated circuit in which the processor clients are internal to that integrated circuit. In this integrated circuit context, there is no significant physical distance between the various processor clients, much less significant differences between respective physical distances associated with the various processor clients, and as a result the alleged motivation is deficient.

Moreover, as indicated above, Ng deals with communications between server devices and storage devices in a SAN, and Sato deals with interconnecting processor elements of a parallel

processor system or supercomputer for purposes of load testing. These are entirely different systems, and there does not appear to be any need whatsoever for the type of load testing disclosed in Sato within the SAN of Ng.

Appellants therefore respectfully submit that the statements proffered by the Examiner fail to provide sufficient objective motivation for the combination and, rather, are conclusory statements of the sort rejected by both the Federal Circuit and the U.S. Supreme Court. See *KSR v. Teleflex*, 127 S. Ct. 1727, 1741 (2007), quoting *In re Kahn*, 441 F. 3d 977, 988 (Fed. Cir. 2006) (“[R]ejections on obviousness grounds cannot be sustained by mere conclusory statements; instead, there must be some articulated reasoning with some rational underpinning to support the legal conclusion of obviousness.”)

Independent claim 20 is believed allowable for reasons similar to those identified above with regard to claim 1.

Dependent claims 2-10 and 16 are believed allowable at least by virtue of their dependence from claim 1.

Claim 11

In addition to being allowable by virtue of its dependency from independent claim 1, dependent claim 11 recites separately patentable subject matter. Specifically, dependent claim 11 includes a limitation wherein addresses are allocated to multiple memory instances associated with the given processor client in order of decreasing memory instance size. An illustrative embodiment is described in the specification at page 11, lines 9-12, with reference to FIG. 4A, in which the biggest memory instance is allocated first, while the smallest memory instance is allocated last.

In formulating the present rejection of claim 11 in the final Office Action at page 6, first paragraph, the Examiner argues that this limitation is disclosed by Ng at paragraph [0035], which states that “[f]ree pools represent the available storage from the storage pool before allocation. A system administrator or software agent divides the storage into logical portions when deciding how to partition the free pools into allocated pools of storage. . . . VLUN or Virtual LUN is storage carved from a virtualized pool of storage that the VSX devices export to an initiator port.

For example, the VLUN is [sic] can be created by concatenating multiple physical LUNs or can use only a single slice cut from a single LUN of RAIDs or JBODs.”

Appellants respectfully submit that there is simply no teaching or even suggestion in Ng directed to the limitation of claim 11 wherein addresses are allocated to multiple memory instances associated with a given processor client in order of decreasing memory instance size. Indeed, there is no teaching or suggestion regarding any consideration of memory instance size in allocating addresses to multiple memory instances associated with a given processor client. Sato fails to supplement the deficient teachings of Ng, and hence claim 11 is allowable over the combination of Ng and Sato.

Claim 12

In addition to being allowable by virtue of its dependency from independent claim 1, dependent claim 12 recites separately patentable subject matter. Specifically, dependent claim 12 includes a limitation wherein multiple memory instances associated with the given processor client have different sizes which are related to one another as multiples of two.

In an illustrative embodiment described in the specification at, for example, page 10, line 24, to page 11, line 4, the sizes of the internal memory instances can vary from 32 KB to 1 MB in multiples of two. With reference to the table in FIG. 4A, a given client in this example may be assigned the nine memory instances denoted mem1, mem2, . . . mem9. Mem1 and mem2 each have a size of 1 MB, mem3 and mem4 each have a size of 512 KB, mem5 has a size of 256 KB, mem6 and mem7 each have a size of 128 KB, mem8 has a size of 64 KB, and mem9 has a size of 32 KB.

In formulating the present rejection of claim 12 in the final Office Action at page 6, second paragraph, the Examiner argues that this limitation is disclosed by Ng at paragraph [0035], which states that “[f]ree pools represent the available storage from the storage pool before allocation. A system administrator or software agent divides the storage into logical portions when deciding how to partition the free pools into allocated pools of storage. . . . VLUN or Virtual LUN is storage carved from a virtualized pool of storage that the VSX devices export

to an initiator port. For example, the VLUN is [sic] can be created by concatenating multiple physical LUNs or can use only a single slice cut from a single LUN of RAIDs or JBODs.”

Appellants respectfully submit that there is simply no teaching or even suggestion in Ng directed to the limitation of claim 12 wherein multiple memory instances associated with the given processor client have different sizes which are related to one another as multiples of two. Indeed, there is no teaching or suggestion regarding the relative sizes, or even the absolute sizes, of multiple memory instances associated with a given processor client. Sato fails to supplement the deficient teachings of Ng, and hence claim 12 is allowable over the combination of Ng and Sato.

Claim 13

In addition to being allowable by virtue of its dependency from independent claim 1, dependent claim 13 recites separately patentable subject matter. Specifically, dependent claim 13 includes limitations wherein a different set of mask bits is associated with each of a plurality of different memory instance sizes, and a different address decoder value is associated with each of the plurality of memory instances.

In an illustrative embodiment described in the specification at, for example, page 11, lines 11-27, a different set of mask bits is associated with each of the different memory instance sizes. This is shown in the table of FIG. 4B, which lists the memory sizes and the particular set of mask bits associated with each memory size. In this example, which utilizes a 20-bit address denoted as bits [19:0], the mask bits comprise a seven-bit value denoted as bits [6:0]. Thus, the mask bits associated with the 1 MB memory instance size are 0011111, the mask bits associated with the 512 KB memory instance size are 0001111, the mask bits associated with the 256 KB memory instance size are 0000111, and so on.

The table in FIG. 4A indicates that a different address decoder value is associated with each of the nine memory instances mem1 through mem9. Once the memory instances for the given client are organized in decreasing size as described above with regard to claim 11, with the biggest memory instance starting with the 0th address, the upper seven bits of the memory instance start address are used as the address decoder value for the biggest memory instance. So

in the FIG. 4A example, mem1 has start address 00000h so the corresponding address decoder value is the seven-bit value 0000000b, where “b” denotes binary notation. For mem2 the start address is 40000h, so its corresponding address decoder value is the seven-bit value 0100000b. For mem9 the start address is E4000h, so its corresponding address decoder value is the seven-bit value 1110010b.

In formulating the present rejection of claim 13 in the final Office Action at page 6, third paragraph, the Examiner argues that these limitations are disclosed by Ng at paragraphs [0056] and [0057]. The relied-upon portion of Ng teaches, with reference to step 602 in FIG. 6, that as part of an operation for setting up flow control to monitor and limit traffic carrying data and commands over the various ports and storage devices, an administrator or agent identifies a port-storage identifier and the corresponding initiator-target-LUN (ITL) combination along with the associated data type to be monitored. It should be noted that, as described in Ng at paragraphs [0024] to [0025] and [0039] to [0041], an initiator-target-LUN (ITL) combination, more generally referred to as a port-storage identifier, identifies a given path from an initiator port (i.e., a port on a host server) through a target port (i.e., a port on the VSX device), and onto a storage device port.

Appellants respectfully submit that there is simply no teaching or even suggestion in Ng directed to the limitations of claim 13 wherein a different set of mask bits is associated with each of a plurality of different memory instance sizes, and a different address decoder value is associated with each of the plurality of memory instances. There is no teaching or suggestion regarding either associating a different set of mask bits is associated with each of a plurality of different memory instance sizes or associating a different address decoder value with each of the plurality of memory instances. Indeed, Ng fails to even teach or suggest an arrangement wherein a memory instance has a set of mask bits and an address decoder value associated therewith. Sato fails to supplement the deficient teachings of Ng, and hence claim 13 is allowable over the combination of Ng and Sato.

Claim 14

In addition to being allowable by virtue of its dependency from claims 1 and 13, dependent claim 14 recites separately patentable subject matter. Specifically, dependent claim 14 includes a limitation wherein address decoding logic applies the mask bits for a given memory instance to an incoming address from the given processor client, and compares the result to the address decoder value for the given memory instance to determine if the incoming address is directed to an address in the given memory instance.

In an illustrative embodiment described in the specification at, for example, page 12, lines 3-12, the address decoding logic applies the mask bits to the upper seven bits of an incoming address, and compares the result to an address decoder value stored in a configuration register to see if there is an address hit for a particular one of the memory instances. In the event of an address hit, the incoming address can be passed as is to the appropriate memory instance.

In formulating the present rejection of claim 14 in the final Office Action at page 6, fourth paragraph, the Examiner argues that this limitation is disclosed by Ng at paragraphs [0056] and [0057]. The relied-upon portion of Ng teaches, with reference to step 602 in FIG. 6, that as part of an operation for setting up flow control to monitor and limit traffic carrying data and commands over the various ports and storage devices, an administrator or agent identifies a port-storage identifier and the corresponding initiator-target-LUN (ITL) combination along with the associated data type to be monitored. It should be noted that, as described in Ng at paragraphs [0024] to [0025] and [0039] to [0041], an initiator-target-LUN (ITL) combination, more generally referred to as a port-storage identifier, identifies a given path from an initiator port (i.e., a port on a host server) through a target port (i.e., a port on the VSX device), and onto a storage device port.

There is simply no teaching or even suggestion in Ng directed to the limitations of claim 14 wherein address decoding logic applies the mask bits for a given memory instance to an incoming address from the given processor client, and compares the result to the address decoder value for the given memory instance to determine if the incoming address is directed to an address in the given memory instance. There is no teaching or suggestion regarding either applying the mask bits for a given memory instance to an incoming address from the given

processor client or comparing the result to the address decoder value for the given memory instance to determine if the incoming address is directed to an address in the given memory instance. Indeed, Ng fails to even teach or suggest an arrangement wherein a memory instance has a set of mask bits and an address decoder value associated therewith. Sato fails to supplement the deficient teachings of Ng, and hence claim 14 is allowable over the combination of Ng and Sato.

Claim 15

In addition to being allowable by virtue of its dependency from claims 1, 13, and 14, dependent claim 15 recites separately patentable subject matter. Specifically, dependent claim 15 includes a limitation wherein a decoded address is considered valid for the given processor client only if a master client identifier stored for the given memory instance specifies the given processor client.

In an illustrative embodiment described in the specification at, for example, page 13, lines 8-22, a stored five-bit value identifies a “master” client for each memory instance. In this example, it is assumed that there are no more than 20 distinct clients, denoted CL1 through CL20, with each client having a particular five-bit identifier, namely 00001 for CL1, . . . , 10100 for CL20. The master client identifier specifies which of the clients, if any, has been assigned to the particular memory instance. This allows the address decoder logic to determine the appropriate client whose access request address is validly decoded for the particular memory instance, so that any other such access requests from other clients that are directed to the particular memory instance can be ignored. Thus, a decoded address is considered valid for a given client only if the master client identifier stored for the particular memory instance specifies the given client. A value of 00000 in this implementation indicates that there is no master client for the particular memory instance.

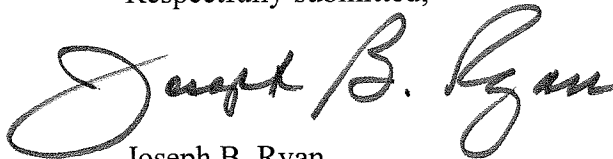
In formulating the present rejection of claim 15 in the final Office Action at page 6, fourth paragraph, the Examiner argues that this limitation is disclosed by Ng at paragraphs [0056] and [0057]. The relied-upon portion of Ng teaches, with reference to step 602 in FIG. 6, that as part of an operation for setting up flow control to monitor and limit traffic carrying data

and commands over the various ports and storage devices, an administrator or agent identifies a port-storage identifier and the corresponding initiator-target-LUN (ITL) combination along with the associated data type to be monitored. It should be noted that, as described in Ng at paragraphs [0024] to [0025] and [0039] to [0041], an initiator-target-LUN (ITL) combination, more generally referred to as a port-storage identifier, identifies a given path from an initiator port (i.e., a port on a host server) through a target port (i.e., a port on the VSX device), and onto a storage device port.

There is simply no teaching or even suggestion in Ng directed to the limitations of claim 15 wherein a decoded address is considered valid for the given processor client only if a master client identifier stored for the given memory instance specifies the given processor client. Indeed, Ng fails to even teach or suggest an arrangement wherein a master client identifier is stored for a given memory instance. Sato fails to supplement the deficient teachings of Ng, and hence claim 15 is allowable over the combination of Ng and Sato.

In view of the above, Appellants believe that claims 1-16 and 20 are in condition for allowance, and respectfully request reversal of the §103(a) rejection.

Respectfully submitted,

A handwritten signature in black ink, reading "Joseph B. Ryan". The signature is fluid and cursive, with the first name "Joseph" being more prominent and the last name "Ryan" following in a similar style.

Date: February 9, 2009

Joseph B. Ryan
Attorney for Appellant(s)
Reg. No. 37,922
Ryan, Mason & Lewis, LLP
90 Forest Avenue
Locust Valley, NY 11560
(516) 759-7517

CLAIMS APPENDIX

1. A network processor integrated circuit comprising:

a plurality of processor clients internal to the network processor integrated circuit;

an internal memory having a plurality of memory instances; and

an internal memory controller for controlling access of the plurality of processor clients to the plurality of memory instances, the internal memory controller comprising a configurable switching element;

the configurable switching element being connectable between the plurality of processor clients and the plurality of memory instances and being operative to control access of particular ones of the plurality of processor clients to particular ones of the plurality of memory instances;

wherein the configurable switching element is configurable to connect any one of at least a subset of the plurality of processor clients to each of at least a subset of the plurality of memory instances,

such that in a first selectable configuration of the configurable switching element, a given one of the processor clients is permitted to access a first set of memory instances comprising one or more of the plurality of memory instances, and in a second selectable configuration of the configurable switching element, the given processor client is permitted to access a second set of memory instances comprising one or more of the plurality of memory instances, the second set being different than the first set.

2. The network processor integrated circuit of claim 1 wherein the configurable switching element comprises a configurable crossbar having a first set of ports coupled to the plurality of processor clients and a second set of ports coupled to the plurality of memory instances.

3. The network processor integrated circuit of claim 1 wherein the memory controller further comprises control circuitry operative to control selection of a particular configuration for the configurable switching element.

4. The network processor integrated circuit of claim 3 wherein the control circuitry further comprises an address control circuit.

5. The network processor integrated circuit of claim 3 wherein the control circuitry further comprises a data multiplexing control circuit.

6. The network processor integrated circuit of claim 1 wherein the internal memory controller further comprises a configuration interface providing an interface between the configurable switching element and a configuration source external to the memory controller, the external configuration source providing to the memory controller information utilizable to control selection of a particular configuration for the configurable switching element.

7. The network processor integrated circuit of claim 1 wherein the plurality of processor clients comprises N processor clients, and the plurality of memory instances comprises M memory instances, where N need not be equal to M.

8. The network processor integrated circuit of claim 7 wherein N is less than M.

9. The network processor integrated circuit of claim 1 wherein the configurable switching element is configurable to connect any one of the plurality of processor clients to any set of memory instances comprising one or more of the plurality of memory instances.

10. The network processor integrated circuit of claim 1 wherein for a given configuration of the configurable switching element, each of at least a subset of the memory instances has one and only one of the processor clients assigned to it.

11. The network processor integrated circuit of claim 1 wherein addresses are allocated to multiple memory instances associated with the given processor client in order of decreasing memory instance size.

12. The network processor integrated circuit of claim 1 wherein multiple memory instances associated with the given processor client have different sizes which are related to one another as multiples of two.

13. The network processor integrated circuit of claim 1 wherein a different set of mask bits is associated with each of a plurality of different memory instance sizes, and a different address decoder value is associated with each of the plurality of memory instances.

14. The network processor integrated circuit of claim 13 wherein address decoding logic applies the mask bits for a given memory instance to an incoming address from the given processor client, and compares the result to the address decoder value for the given memory instance to determine if the incoming address is directed to an address in the given memory instance.

15. The network processor integrated circuit of claim 14 wherein a decoded address is considered valid for the given processor client only if a master client identifier stored for the given memory instance specifies the given processor client.

16. The network processor integrated circuit of claim 1 wherein the network processor integrated circuit is configured to provide an interface for communication of protocol data units between a network and a switch fabric.

17. (Canceled)

18. (Canceled)

19. (Canceled)

20. A method for use in a network processor integrated circuit for controlling access of a plurality of processor clients internal to the network processor integrated circuit to a plurality of memory instances of an internal memory of the network processor integrated circuit, the method comprising the steps of:

providing within the network processor integrated circuit an internal memory controller comprising a configurable switching element;

the configurable switching element being connectable between the plurality of processor clients and the plurality of memory instances and being operative to control access of particular ones of the plurality of processor clients to particular ones of the plurality of memory instances;

wherein the configurable switching element is configurable to connect any one of at least a subset of the plurality of processor clients to each of at least a subset of the plurality of memory instances; and

selecting one of at least a first selectable configuration and a second selectable configuration of the configurable switching element, wherein in the first selectable configuration a given one of the processor clients is permitted to access a first set of memory instances comprising one or more of the plurality of memory instances, and in the second selectable configuration the given processor client is permitted to access a second set of memory instances comprising one or more of the plurality of memory instances, the second set being different than the first set.

EVIDENCE APPENDIX

None.

RELATED PROCEEDINGS APPENDIX

None.